A Rule-based Question Answering System for Reading Comprehension Tests

Ellen Riloff and **Michael Thelen**
Department of Computer Science
University of Utah
Salt Lake City, Utah 84112
{riloff,thelenm}@cs.utah.edu

Abstract

We have developed a rule-based system, Quarc, that can read a short story and find the sentence in the story that best answers a given question. Quarc uses heuristic rules that look for lexical and semantic clues in the question and the story. We have tested Quarc on reading comprehension tests typically given to children in grades 3-6. Overall, Quarc found the correct sentence 40% of the time, which is encouraging given the simplicity of its rules.

1 Introduction

In the United States, we evaluate the reading ability of children by giving them reading comprehension tests. These test typically consist of a short story followed by questions. Presumably, the tests are designed so that the reader must understand important aspects of the story to answer the questions correctly. For this reason, we believe that reading comprehension tests can be a valuable tool to assess the state of the art in natural language understanding.

These tests are especially challenging because they can discuss virtually any topic. Consequently, broad-coverage natural language processing (NLP) techniques must be used. But the reading comprehension tests also require semantic understanding, which is difficult to achieve with broad-coverage techniques.

We have developed a system called Quarc that "takes" reading comprehension tests. Given a story and a question, Quarc finds the sentence in the story that best answers the question. Quarc does not use deep language understanding or sophisticated techniques, yet it achieved 40% accuracy in our experiments. Quarc uses hand-crafted heuristic rules that look for lexical and semantic clues in the question and the story. In the next section, we describe the reading comprehension tests. In the following sections, we describe the rules used by Quarc and present experimental results.

2 Reading Comprehension Tests

Figure 1 shows an example of a reading comprehension test from Remedia Publications. Each test is followed by five "WH" questions: WHO, WHAT, WHEN, WHERE, and WHY.[1] The answers to the questions typically refer to a string in the text, such as a name or description, which can range in length from a single noun phrase to an entire clause or sentence. The answers to WHEN and WHERE questions are also sometimes inferred from the dateline of the story. For example, (EGYPT, 1951) contains the answer to the WHEN question in Figure 1.

Ideally, a natural language processing system would produce the exact answer to a question. Identifying the precise boundaries of the answer can be tricky, however. We will focus on the somewhat easier task of identifying the sentence that contains the answer to a question.

3 A Rule-based System for Question Answering

Quarc (QUestion Answering for Reading Comprehension) is a rule-based system that uses lexical and semantic heuristics to look for evidence that a sentence contains the answer to a question. Each type of WH question looks for different types of answers, so Quarc uses a separate set of rules for each question type (WHO, WHAT, WHEN, WHERE, WHY).

Given a question and a story, Quarc parses the question and all of the sentences in the story using our partial parser Sundance. Much of

[1] There is also a lone HOW question in the data set, but we ignored it.

Tomb Keeps Its Secrets

(EGYPT, 1951) - A tomb was found this year. It was a tomb built for a king. The king lived more than 4,000 years ago. His home was in Egypt.

For years, no one saw the tomb. It was carved deep in rock. The wind blew sand over the top and hid it. Then a team of diggers came along. Their job was to search for hidden treasures.

What they found thrilled them. Jewels and gold were found in the tomb. The king's treasures were buried inside 132 rooms.

The men opened a 10-foot-thick door. It was 130 feet below the earth. Using torches, they saw a case. "It must contain the king's mummy!" they said. A mummy is a body wrapped in sheets.

With great care, the case was removed. It was taken to a safe place to be opened. For two hours, workers tried to lift the lid. At last, they got it off.

Inside they saw ... nothing! The case was empty. No one knows where the body is hidden. A new mystery has begun.

1. Who was supposed to be buried in the tomb?
2. What is a mummy?
3. When did this story happen?
4. Where was the 10-foot-thick door found?
5. Why was the body gone?

Figure 1: Sample Reading Comprehension Test

the syntactic analysis is not used, but Quarc does use the morphological analysis, part-of-speech tagging, semantic class tagging, and entity recognition. The rules are applied to each sentence in the story, as well as the title of the story, with the exception that the title is not considered for WHY questions. The dateline is also a possible answer for WHEN and WHERE questions and is evaluated using a special set of dateline rules.

Each rule awards a certain number of points to a sentence. After all of the rules have been applied, the sentence (or dateline) that obtains the highest score is returned as the answer.

All of the question types share a common WordMatch function, which counts the number of words that appear in both the question and the sentence being considered. The WordMatch function first strips a sentence of stopwords[2], and then matches the remaining words against the words in the question. Two words match if they share the same morphological root. Verbs seem to be especially important for recognizing that a question and sentence are related, so verb matches are weighted more heavily than non-verb matches. Matching verbs are awarded 6 points each and other matching words are awarded 3 points each.

The other rules used by Quarc look for a variety of clues. Lexical clues look for specific words or phrases. Unless a rule indicates otherwise, words are compared using their morphological roots. Some rules can be satisfied by any of several lexical items; these rules are written using set notation (e.g., {*yesterday,today,tomorrow*}). Some rules also look for semantic classes, which we will write in upper case (e.g., HUMAN). Our parser uses a dictionary and a semantic hierarchy, so that words can be defined with semantic classes. The semantic classes used by Quarc are shown below, along with a description of the words assigned to each class.

HUMAN: 2608 words,[3] including common first names, last names, titles such as "Dr." and "Mrs.", and about 600 occupation words acquired from WordNet (Miller, 1990).

LOCATION: 344 words, including 204 country names and the 50 United States.

MONTH: the 12 months of the year.

TIME: 667 words, 600 of which are enumerated years from 1400-1999. The others are general time expressions, including the 12 months of the year.

[2]We used a stopword list containing 41 words, mostly prepositions, pronouns, and auxiliary verbs.

[3]About 2000 words came from the Social Security Administration's list of the top 1000 names for each gender in 1998: www.ssa.gov/OACT/NOTES/note139/1998/top1000in98.html.

Our parser also recognizes two types of semantic entities: proper nouns and names. A PROPER_NOUN is defined as a noun phrase in which all words are capitalized. A NAME is defined as a PROPER_NOUN that contains at least one HUMAN word.

Each rule awards a specific number of points to a sentence, depending on how strongly the rule believes that it found the answer. A rule can assign four possible point values: **clue** (+3), **good_clue** (+4), **confident** (+6), and **slam_dunk** (+20). These point values were based on our intuitions and worked well empirically, but they are not well justified. The main purpose of these values is to assess the relative importance of each clue.

Figure 2 shows the WHO rules, which use three fairly general heuristics as well as the WordMatch function (rule #1). If the question (Q) does not contain any names, then rules #2 and #3 assume that the question is looking for a name. Rule #2 rewards sentences that contain a recognized NAME, and rule #3 rewards sentences that contain the word "name". Rule #4 awards points to all sentences that contain either a name or a reference to a human (often an occupation, such as "writer"). Note that more than one rule can apply to a sentence, in which case the sentence is awarded points by all of the rules that applied.

1. Score(S) += WordMatch(Q,S)
2. If ¬ contains(Q,NAME) and
   contains(S,NAME)
   Then Score(S) += **confident**
3. If ¬ contains(Q,NAME) and
   contains(S,*name*)
   Then Score(S) += **good_clue**
4. If contains(S,{NAME,HUMAN})
   Then Score(S) += **good_clue**

Figure 2: WHO Rules

The WHAT questions were the most difficult to handle because they sought an amazing variety of answers. But Figure 3 shows a few specific rules that worked reasonably well. Rule #1 is the generic word matching function shared by all question types. Rule #2 rewards sentences that contain a date expression if the question contains a month of the year. This rule handles questions that ask what occurred on a specific date. We also noticed several "what kind?" questions, which looked for a description of an object. Rule #3 addresses these questions by rewarding sentences that contain the word "call" or "from" (e.g., "It is called..." or "It is made from ..."). Rule #4 looks for words associated with names in both the question and sentence. Rule #5 is very specific and recognizes questions that contain phrases such as "name of <x>" or "name for <x>". Any sentence that contains a proper noun whose head noun matches <x> will be highly rewarded. For example, the question "What is the name of the creek?" is answered by a sentence that contains the noun phrase "Pigeon Creek".

1. Score(S) += WordMatch(Q,S)
2. If contains(Q,MONTH) and
   contains(S,{*today*,*yesterday*,
   tomorrow,*last night*})
   Then Score(S) += **clue**
3. If contains(Q,*kind*) and
   contains(S,{*call*,*from*})
   Then Score(S) += **good_clue**
4. If contains(Q,*name*) and
   contains(S,{*name*,*call*,*known*})
   Then Score += **slam_dunk**
5. If contains(Q,*name*+PP) and
   contains(S,PROPER_NOUN) and
   contains(PROPER_NOUN,head(PP))
   Then Score(S) += **slam_dunk**

Figure 3: WHAT Rules

The rule set for WHEN questions, shown in Figure 4, is the only rule set that does not apply the WordMatch function to every sentence in the story. WHEN questions almost always require a TIME expression, so sentences that do not contain a TIME expression are only considered in special cases. Rule #1 rewards all sentences that contain a TIME expression with **good_clue** points as well as WordMatch points. The remaining rules look for specific words that suggest a duration of time. Rule #3 is interesting because it recognizes that certain verbs ("begin", "start") can be indicative of time even when no specific time is mentioned.

The WHERE questions almost always look for specific locations, so the WHERE rules are very focused. Rule #1 applies the general word matching function and Rule #2 looks for sen-

1. If contains(S,TIME)
   Then Score(S) += **good_clue**
   Score(S) += WordMatch(Q,S)
2. If contains(Q,*the last*) and
   contains(S,{*first*,*last*,*since*,*ago*})
   Then Score(S) += **slam_dunk**
3. If contains(Q,{*start*,*begin*}) and
   contains(S,{*start*,*begin*,*since*,*year*})
   Then Score(S) += **slam_dunk**

Figure 4: WHEN Rules

1. Score(S) += WordMatch(Q,S)
2. If contains(S,LocationPrep)
   Then Score(S) += **good_clue**
3. If contains(S, LOCATION)
   Then Score(S) += **confident**

Figure 5: WHERE Rules

tences with a location preposition. Quarc recognizes 21 prepositions as being associated with locations, such as "in", "at", "near", and "inside". Rule #3 looks for sentences that contain a word belonging to the LOCATION semantic class.

WHY questions are handled differently than other questions. The WHY rules are based on the observation that the answer to a WHY question often appears immediately before or immediately after the sentence that most closely matches the question. We believe that this is due to the causal nature of WHY questions.

First, all sentences are assigned a score using the WordMatch function. Then the sentences with the top score are isolated. We will refer to these sentences as BEST. Every sentence score is then reinitialized to zero and the WHY rules, shown in Figure 6, are applied to every sentence in the story.

Rule #1 rewards all sentences that produced the best WordMatch score because they are plausible candidates. Rule #2 rewards sentences that immediately precede a best WordMatch sentence, and Rule #3 rewards sentences that immediately follow a best WordMatch sentence. Rule #3 gives a higher score than Rules #1 and #2 because we observed that WHY answers are somewhat more likely to follow the best WordMatch sentence. Finally, Rule #4 rewards sentences that contain the word "want"

1. If S ϵ BEST
   Then Score(S) += **clue**
2. If S immed. precedes member of BEST
   Then Score(S) += **clue**
3. If S immed. follows member of BEST
   Then Score(S) += **good_clue**
4. If contains(S,*want*)
   Then Score(S) += **good_clue**
5. If contains(S,{*so*,*because*})
   Then Score(S) += **good_clue**

Figure 6: WHY Rules

and Rule #5 rewards sentences that contain the word "so" or "because". These words are indicative of intentions, explanations, and justifications.

The answers to WHEN and WHERE questions are frequently found in the dateline rather than the story itself, so Quarc also considers the dateline as a possible answer. Figure 7 shows the dateline rules, which are used for both WHEN and WHERE questions. The words "happen" and "take place" suggest that the dateline may be the best answer (rules #1 and #2). We also found that that the words "this" and "story" were strong indicators that the dateline is the best answer (rules #3 and #4). We found several sentences of the form "When did this happen?" or "When did this take place?". The verbs alone are not sufficient to be slam dunks because they often have a specific subject (e.g., "When did the surprise happen?") that refers back to a sentence in the story. But when the words "story" or "this" appear, the question seems to be referring to the story in its entirety and the dateline is the best answer.

1. If contains(Q,*happen*)
   Then Score(DATELINE) += **good_clue**
2. If contains(Q,*take*) and
   contains(Q,*place*)
   Then Score(DATELINE) += **good_clue**
3. If contains(Q,*this*)
   Then Score(DATELINE) += **slam_dunk**
4. If contains(Q,*story*)
   Then Score(DATELINE) += **slam_dunk**

Figure 7: Dateline Rules

After all the rules have been applied to every sentence in the story, the sentence (or dateline) with the highest score is returned as the best answer. In the event of a tie, a WHY question chooses the sentence that appears latest in the story, and all other question types choose the sentence that appears earliest in the story. If no sentence receives a positive score, then WHEN and WHERE questions return the dateline as a default, WHY questions return the last sentence in the story, and all other questions return the first sentence in the story.

4 Experimental Results

We evaluated Quarc on the same data set that was used to evaluate the DeepRead reading comprehension system (Hirschman et al., 1999). This data set contains 115 reading comprehension tests, 55 of which were used for development and 60 of which were reserved for testing purposes. We also used the answer keys created by the DeepRead developers (Hirschman et al., 1999). The **HumSent** answers are sentences that a human judged to be the best answer for each question. The **AutSent** answers are generated automatically by determining which sentence contains the highest percentage of words in the published answer key, excluding stopwords. We focused on obtaining the best possible **HumSent** score because we believed that humans were more reliable than the automatic word-counting routine.

Table 1 shows Quarc's results for each type of question as well as its overall results. Quarc achieved 40% **HumSent** accuracy overall, but the accuracy varied substantially across question types. Quarc performed the best on WHEN questions, achieving 55% accuracy, and performed the worst on WHAT and WHY questions, reaching only 28% accuracy.

Quarc's rules use a variety of knowledge sources, so we ran a set of experiments to evaluate the contribution of each type of knowledge. Figure 8 shows the results of these experiments, based on the **HumSent** answer keys. First, we evaluated the performance of Quarc's WordMatch function all by itself, giving equal weight to verbs and non-verbs. The WordMatch function alone, shown as Word on the graph, produced 27% accuracy. When we gave verbs twice as much weight as non-verbs (*+Verb*), overall accuracy improved to 28%. Interestingly, giving extra weight to verbs improved the WHO and WHAT questions, but hurt the WHEN and WHERE questions. These results suggest that verbs should be weighted more heavily only for certain question types, even though we

WHO		
HumSent:	0.41	(24/59)
AutSent:	0.49	(29/59)
WHAT		
HumSent:	0.28	(17/61)
AutSent:	0.31	(19/61)
WHEN		
HumSent:	0.55	(33/60)
AutSent:	0.28	(17/60)
WHERE		
HumSent:	0.47	(28/60)
AutSent:	0.48	(29/60)
WHY		
HumSent:	0.28	(17/60)
AutSent:	0.27	(16/60)
OVERALL		
HumSent:	0.40	(119/300)
AutSent:	0.37	(110/300)

Table 1: Overall Results

Next, we wanted to see how much effect the semantic classes had on performance, so we added the rules that use semantic classes. Only the WHO, WHEN, and WHAT question types had such rules, and performance improved on those question types (*+Sem*). We then added the dateline rules for the WHEN and WHERE questions, and added the WHY rules that reward the sentences immediately preceding and following the best WordMatch sentence (rules #1-3 in Figure 6). Figure 8 shows that these additions (*+Why/Dateline*) also improved results for all three question types.

Finally, we added the remaining rules that look for specific words and phrases. The final version of Quarc achieved 40% **HumSent** accuracy, which compares favorably with DeepRead's results (36% **HumSent** accuracy). Furthermore, DeepRead's best results used hand-tagged named entity recognition and hand-tagged coreference resolution. Quarc did not rely on any hand-tagging and did not perform any coreference reslution.

We also ran an experiment to evaluate the quality of Quarc's tie-breaking procedure, which was described at the end of Section 3. When

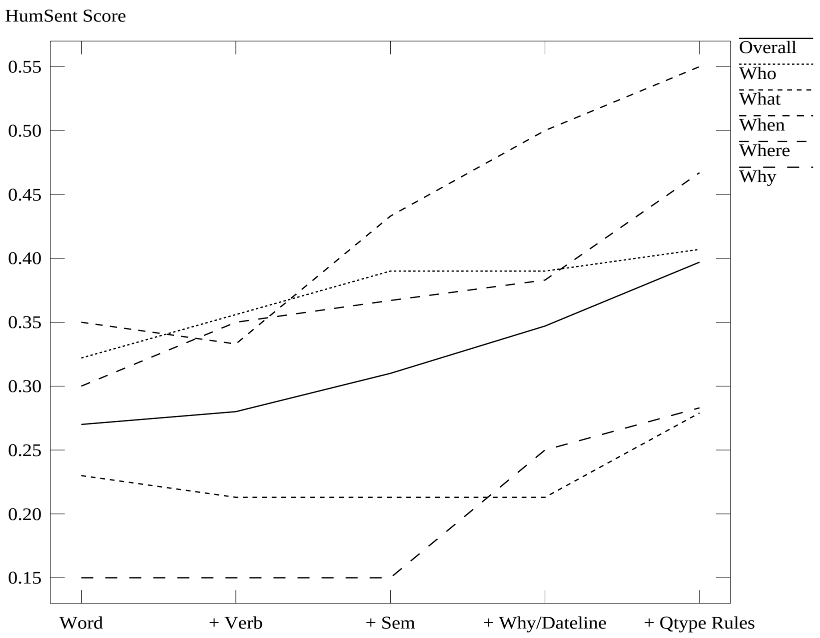

Figure 8: Experimental Results

more than one sentence is tied with the best score, Quarc selects the sentence that appears earliest in the story, except for WHY questions when Quarc chooses the sentence appearing latest in the story. Table 2 shows the results of removing this tie-breaking procedure, so that Quarc is allowed to output all sentences that received the top score. These results represent an upper bound on performance if Quarc had a perfect tie-breaking mechanism.

Table 2 shows that Quarc's performance on WHAT, WHEN, and WHY questions improved by several percentage points, but performance on WHO and WHERE questions was basically the same. Overall, Quarc was able to identify 46% of the correct sentences by generating 1.75 hypotheses per question on average. These results suggest that a better tie-breaking procedure could substantially improve Quarc's performance by choosing between the top two or three candidates more intelligently.

5 Lessons Learned

Quarc's rules were devised by hand after experimenting with the 55 reading comprehension tests in the development set. These simple rules are probably not adequate to handle other types of question-answering tasks, but this exercise gave us some insights into the problem.

First, semantic classes were extremely useful for WHO, WHEN, and WHERE questions because they look for descriptions of people, dates, and locations. Second, WHY questions are concerned with causal information, and we discovered several keywords that were useful for identifying intentions, explanations, and justifications. A better understanding of causal relationships and discourse structure would undoubtedly be very helpful. Finally, WHAT questions were the most difficult because they sought a staggering variety of answers. The only general pattern that we discovered was that WHAT questions often look for a description of an event or an object.

Reading comprehension tests are a wonderful testbed for research in natural language processing because they require broad-coverage techniques and semantic knowledge. In the future, we plan to incorporate coreference resolution, which seems to be very important for this task. We also plan to experiment with techniques that acquire semantic knowledge automatically (e.g., (Riloff and Shepherd, 1997; Roark and Charniak, 1998)) to generate bigger and better semantic lexicons.

WHO		
HumSent:	0.42	(25/59)
AutSent:	0.53	(31/59)
Avg # answers:	1.27	
WHAT		
HumSent:	0.44	(27/61)
AutSent:	0.49	(30/61)
Avg # answers:	2.84	
WHEN		
HumSent:	0.62	(37/60)
AutSent:	0.32	(19/60)
Avg # answers:	1.45	
WHERE		
HumSent:	0.48	(29/60)
AutSent:	0.48	(29/60)
Avg # answers:	1.33	
WHY		
HumSent:	0.33	(20/60)
AutSent:	0.30	(18/60)
Avg # answers:	1.82	
OVERALL		
HumSent:	0.46	(138/300)
AutSent:	0.42	(127/300)
Avg # answers:	1.75	

Table 2: Generating multiple answers

6 Acknowledgments

This research is supported in part by the National Science Foundation under grant IRI-9704240.

References

L. Hirschman, M. Light, E. Breck, and J. Burger. 1999. Deep Read: A Reading Comprehension System. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

G. Miller. 1990. Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4).

E. Riloff and J. Shepherd. 1997. A Corpus-Based Approach for Building Semantic Lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124.

B. Roark and E. Charniak. 1998. Noun-phrase Co-occurrence Statistics for Semi-automatic Semantic Lexicon Construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 1110–1116.